

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS:

1. (Currently Amended) A computer system, comprising:
 - a display device on which windows and other graphical user interface elements are displayed;
 - a cursor control device for positioning a cursor displayed on said display device relative to said user interface elements; and
 - a user interface which controls the appearance of said cursor to display a first image when said cursor is positioned over a user interface element that can be immediately accessed by a user, a second image when said cursor is positioned over a ~~user interface element~~ window associated with a foreground ~~process~~ application that is in a busy state that prevents it from being currently accessed, and a third image when said cursor is positioned over a user interface element associated with a background process that is in a busy state;
- wherein the window has a first portion associated with said application and a second portion associated with said user interface, and wherein said second image is only displayed when said cursor is positioned over said first portion of the window.

Claims 2-3 (Canceled)

4. (Original) The computer system of claim 1, wherein said first image comprises a pointer, and each of said second and third images comprise designs that indicate a wait state.

5. (Original) The computer system of claim 4, wherein each of said wait state designs is animated.

6. (Original) The computer system of claim 4, wherein said third image comprises a hybrid of the design of said second image and a pointer.

Claims 7-8 (Canceled)

9. (Original) A method for displaying a cursor in a computer user interface, comprising the steps of:

detecting when the cursor is positioned relative to an object associated with a process;

determining whether the process is in a state where its functionality can be currently accessed or in a busy state;

for a process that is in a busy state, determining whether the process is operating in the foreground or the background; and

selectively displaying a first image for the cursor if the process is busy and operating in the foreground, or a second image for the cursor if the process is busy and operating in the background.

10. (Original) The method of claim 9, further including the step of displaying a third image for the cursor if the process is in said state where its functionality can be currently accessed.

11. (Original) The method of claim 10, wherein the image associated with an object that is currently accessible comprises a pointer, the image associated with a busy foreground process indicates a wait state, and the image associated with a busy background process comprises a hybrid of said other two images.

12. (Original) The method of claim 9, wherein said first image comprises a symbol that represents a wait state, and said second image comprises a combination of said symbol and an indicator that represents accessibility.

13. (Original) The method of claim 9, wherein said foreground process is an application and said associated user interface object is a window.

14. (Original) The method of claim 13, wherein the window has a first portion associated with said application and a second portion associated with said user interface, and wherein the image associated with a foreground process that is currently busy is only displayed when said cursor is positioned over said first portion of the window.

15. (Original) A computer-readable medium containing a program that executes the following steps:

detecting when a cursor of a user interface is positioned relative to an object associated with a process;

determining whether the process is in a state where its functionality can be currently accessed or in a busy state;

for a process that is in a busy state, determining whether the process is operating in the foreground or the background; and

selectively displaying a first image for the cursor if the process is busy and operating in the foreground, or a second image for the cursor if the process is busy and operating in the background.

16. (Original) The computer-readable medium of claim 15, further including the step of displaying a third image for the cursor if the process is in said state where its functionality can be currently accessed.

17. (Original) The computer-readable medium of claim 15, wherein said first image comprises a symbol that represents a wait state, and said second image comprises a combination of said symbol and an indicator that represents accessibility.

18. (Original) The computer-readable medium of claim 17, wherein said indicator comprises a pointer.

19. (Original) A computer system, comprising:

a display device on which a currently running application is represented to a user by a plurality of user interface objects displayed on a desktop;

a cursor control device for positioning a cursor displayed on said display device relative to said user interface objects; and

a user interface which controls the appearance of said cursor to display a first image when said cursor is positioned over a user interface object operating in the foreground that represents an application in a busy state that prevents it from being currently accessed, and a second image when said cursor is positioned over a user interface object operating in the background that represents an application in a busy state that prevents it from being currently accessed.

20. (Original) The computer system of claim 19, wherein one of said plurality of user interface objects that represents a given application comprises a window, and another of said plurality of objects that represents said given application is a member of the group comprising an icon, a minimized window and a button.

21. (Original) A method for displaying a cursor in a computer user interface, comprising the steps of:

representing an application being executed on a computer by means of a plurality of user interface objects displayed on a desktop of the user interface;

detecting when the cursor is positioned over any of said plurality of user interface objects;

determining whether the application represented by the user interface object over which the cursor is positioned is in a busy state or a currently accessible state; and

selectively displaying a first image for the cursor if the application is currently accessible, a second image for the cursor if the application is in the foreground and is in a busy state, or a third image for the cursor if the application is in the background and is in a busy state, while the cursor is positioned over said object.

22. (Original) The method of claim 21, wherein one of said plurality of user interface objects that represents said application comprises a window, and another of said plurality of objects that represents said application is a member of the group comprising an icon, a minimized window and a button.

23. (Currently Amended) A computer readable medium containing a user interface for a computer, said user interface comprising:

at least two different images for a cursor, including a first image which comprises a pointer arrow having a tail, and a second image which comprises a hybrid consisting of a pointer arrow with a graphic in place of said tail, wherein said graphic represents a condition of a process; and

means for normally displaying a cursor with said first image and for switching the display to said second image upon detecting that said cursor is associated with a user interface object that corresponds with said condition.

24. (Currently Amended) The ~~user-interface~~ computer readable medium of claim 23, wherein said condition is a busy state for an application, and said displaying means switches said display upon detecting that the cursor is positioned over a user interface object associated with an application in a busy state.

25. (Currently Amended) The ~~user-interface~~ computer readable medium of claim 23, wherein said condition is the dragging of an object, and said displaying means switches said display upon initiation of a drag operation.

26. (Currently Amended) The ~~user-interface~~ computer readable medium of claim 25, further including a third image comprising a hybrid consisting of a pointer arrow with a graphic in place of said tail that represents a copy operation, and wherein said displaying means switches said display from said second image to said third image upon detecting that the cursor is positioned over a destination object to which the dragged object can be copied.

27. (Currently Amended) The ~~user-interface~~ computer readable medium of claim 26, wherein the graphic of said second image has a first color, and the graphic of said third image has a second, different color.

28. (Currently Amended) The ~~user-interface~~ computer readable medium of claim 26, wherein said graphic of said second image includes a quantitative value that represents a characteristic of the dragged object.

29. (Currently Amended) The ~~user-interface~~ computer readable medium of claim 28, wherein the graphic of said third image also includes said quantitative value.

30. (Original) The ~~user-interface~~ computer readable medium of claim 25, wherein said graphic of said second image includes a quantitative value that represents a characteristic of the dragged object.

31. (Currently Amended) The ~~user-interface~~ computer readable medium of claim 30, wherein said quantitative value indicates the number of objects that are being dragged.

32. (Currently Amended) The ~~user-interface~~ computer readable medium of claim 30, wherein said quantitative value indicates the size of one or more objects being dragged.

33. (Currently Amended) The ~~user-interface~~ computer readable medium of claim 30, wherein said graphic comprises a geometric object, and the size of said geometric object is dynamically varied to accommodate said quantitative value.

34. (Currently Amended) The ~~user-interface~~ computer readable medium of claim 23, wherein said graphic indicates that an object being dragged will be deleted.

35. (Currently Amended) A method for displaying a cursor ~~in a~~ on a display of a computer user interface, comprising the steps of:

normally displaying a cursor on said display with a first image which comprises a pointer arrow having a tail, and

switching the display representation of said cursor on said display to a second image which comprises a hybrid consisting of a pointer arrow with a graphic in place of said tail, wherein said graphic represents a condition of a process, upon detecting that said cursor is associated with a user interface object that corresponds with said condition.

36. (Original) The method of claim 35, wherein said condition is a busy state for an application, and said display of said cursor is switched upon detecting that the cursor is positioned over a user interface object associated with an application in a busy state.

37. (Original) The method of claim 35, wherein said condition is the dragging of an object, and said display of said cursor is switched upon initiation of a drag operation.

38. (Original) The method of claim 37, further including the step of switching said display from said second image to a third image comprising a hybrid consisting of a pointer arrow with a graphic in place of said tail that represents a copy operation, upon detecting that the cursor is positioned over a destination object to which the dragged object can be copied.

39. (Original) The method of claim 38, wherein the graphic of said second image has a first color, and the graphic of said third image has a second, different color.

40. (Original) The method of claim 38, wherein said graphic of said second image includes a quantitative value that represents a characteristic of the dragged object.

41. (Original) The method of claim 40, wherein the graphic of said third image also includes said quantitative value.

42. (Original) The method of claim 37, wherein said graphic of said second image includes a quantitative value that represents a characteristic of the dragged object.

43. (Original) The method of claim 42, wherein said quantitative value indicates the number of objects that are being dragged.

44. (Original) The method of claim 42, wherein said quantitative value indicates the size of one or more objects being dragged.

45. (Original) The method of claim 42, wherein said graphic comprises a geometric object, and further including the step of dynamically varying the size of said geometric object to accommodate said quantitative value.

46. (Original) The method of claim 35, wherein said graphic indicates that an object being dragged will be deleted.

47. (Currently Amended) A method for displaying a cursor ~~in~~ on a display of a computer user-interface, comprising the steps of:

normally displaying a cursor on a display with a first image; and
switching the display representation of said cursor on said display to a second image which comprises a hybrid consisting of a portion of the first image and a graphic, wherein said graphic represents a condition of a process, upon detecting that said cursor is associated with a user interface object that corresponds with said condition.

48. (Original) The method of claim 47, wherein said condition is a busy state for an application, and said display of said cursor is switched upon detecting that the cursor is positioned over a user interface object associated with an application in a busy state.

49. (Original) The method of claim 47, wherein said condition is the dragging of an object, and said display of said cursor is switched upon initiation of a drag operation.

50. (Original) The method of claim 49, further including the step of switching said display from said second image to a third image comprising a hybrid consisting of the portion of the first image with a graphic that represents a copy operation, upon detecting that the cursor is positioned over a destination object to which the dragged object can be copied.

51. (Original) The method of claim 50, wherein the graphic of said second image has a first color, and the graphic of said third image has a second, different color.

52. (Original) The method of claim 50, wherein said graphic of said second image includes a quantitative value that represents a characteristic of the dragged object.

53. (Original) The method of claim 52, wherein the graphic of said third image also includes said quantitative value.

54. (Original) The method of claim 49, wherein said graphic of said second image includes a quantitative value that represents a characteristic of the dragged object.

55. (Original) The method of claim 54, wherein said quantitative value indicates the number of objects that are being dragged.

56. (Original) The method of claim 54, wherein said quantitative value indicates the size of one or more objects being dragged.

57. (Original) The method of claim 54, wherein said graphic comprises a geometric object, and further including the step of dynamically varying the size of said geometric object to accommodate said quantitative value.

58. (Original) The method of claim 47, wherein said graphic indicates that an object being dragged will be deleted.